UNCLASSIFIED

AD NUMBER
AD451231
LIMITATION CHANGES
TO: **Approved for public release; distribution is unlimited.**
FROM: **Distribution authorized to U.S. Gov't. agencies and their contractors; Administrative/Operational Use; OCT 1984. Other requests shall be referred to Defense Advanced Research Projects Agency, 675 North Randolph Street, Arlington, VA 22203-2114.**
AUTHORITY
RAND ltr, 31 Mar 1966

THIS PAGE IS UNCLASSIFIED

ARPA ORDER NO. 189-61

MEMORANDUM
RM-3753-ARPA
OCTOBER 1964

COMPUTER RECOGNITION OF ON-LINE, HAND-WRITTEN CHARACTERS

M. I. Bernstein

PREPARED FOR:

ADVANCED RESEARCH PROJECTS AGENCY

The RAND Corporation
SANTA MONICA · CALIFORNIA

ARPA ORDER NO. 189-61

MEMORANDUM
RM-3753-ARPA
OCTOBER 1964

COMPUTER RECOGNITION OF ON-LINE, HAND-WRITTEN CHARACTERS

M. I. Bernstein

This research is supported by the Advanced Research Projects Agency under Contract No. SD-79. Any views or conclusions contained in this Memorandum should not be interpreted as representing the official opinion or policy of ARPA.

The RAND Corporation
1700 MAIN ST • SANTA MONICA • CALIFORNIA • 90406

PREFACE

This Memorandum details a method for computer recognition of hand-written characters in an on-line environment. The particular device used for the investigation was the RAND Graphic Input Tablet,* though the technique is not uniquely dependent upon the existence of this device. This effort is but one of the facets of an overall attempt to develop techniques which will increase the facility of the man-computer interface.

---

*M. R. Davis and T. O. Ellis, The RAND Tablet: A Man-Machine Graphical Communication Device, The RAND Corporation, RM-4122-ARPA, August 1964.

SUMMARY

This Memorandum discusses a method for recognizing single, hand-written characters using an on-line graphical input device, such as a digitizing pantograph, a light pen, or the RAND Graphic Input Tablet,[1] as the primary information source. Basically, the method consists of filtering and smoothing the input stream to eliminate as much redundancy as possible. Direction of the stylus movement is quantized into one of eight directions, allowing each stroke of a character to be described as a series of connected straight-line segments. By eliminating various measures on the stroke, the description is size-, position-, and rotation-independent. In order to restore some rotational orientation and to discriminate between open, closed, and multi-stroke characters, end-point comparisons are added to the description.

Descriptions used as the basis for recognition are provided by the user in the form of sample hand-written characters associated with the desired output character. The same program which produces a description of an "unknown" character is used to analyze and provide the descriptions from the users' samples.

Recognition, in the original test of the method, is limited to a complete match using a sorted description table. This implies an importance to the order in which things are arranged in the description.

I. INTRODUCTION

In the recent past, several methods for recognizing hand-written characters have been developed.[2,3] The major difference between these approaches and the one described here is one of constraints. In the previous methods, the user has been required to write each character within a fixed, bounded area and around fixed interior points or through internal boundaries. Our goal, however, was to find a method that was independent of position and size and, to a certain extent, rotation, in order that the prospective user could print in as natural a fashion as possible--within the physical limitations of the input hardware available.

Though no attempt is made in this investigation to solve the problem of associating individual strokes with a character or the separation of characters, the author is aware of its existence. On the other hand, this method does not necessarily preclude such endeavors.

Recognition, in the original test of the method, is limited to a complete match using a sorted description table. This implies an importance to the order in which things are arranged in the description.

II. SMOOTHING, FILTERING, AND HYSTERESIS

Given a device which can provide frequent samples of digital positional information (x,y,z) to a computer, the first step in "recognizing" what is being drawn is to extract from this incoming stream a coherent and appropriately filtered track.

We take the position of a stylus on a bounded surface as $\left(x(t),y(t),z(t)\right)$, where x and y are the surface coordinates of the stylus on the surface at time t, and z is a binary coordinate indicating the position of the stylus with respect to the plane of the surface (either on it or not) at time t. We will be concerned in this Memorandum only with those cases in which the stylus is on the surface; therefore, z will not appear any further in our remarks.

The first acceptable point in a stroke is taken as soon as the stylus is on the surface at some time, t_0; thus $\left(x(t_0),y(t_0)\right)$ becomes (x_0,y_0). A candidate (x_c,y_c) for a point in a stroke is given by the following smoothing and filtering algorithm, so long as the stylus remains on the surface:

$$(x_c,y_c) = \left(\frac{x_i-x(t)}{2}, \frac{y_i-y(t)}{2}\right) . \qquad (1)$$

The point is accepted as the next point in the stroke (x_{i+1}, y_{i+1}) if

$$|x_c - x_i| \text{ or } |y_c - y_i| \geq \epsilon , \tag{1a}$$

where ϵ is some arbitrary constant greater than one unit of the pad's raster.

This algorithm serves two purposes: 1) it smoothes out sharp jumps or breaks which occur because of the discrete nature of the pad's raster; and 2) it minimizes the number of points taken for each stroke during periods of slow stylus movement.

III. SEGMENTS

As each new coordinate pair (x_{i+1}, y_{i+1}) is accepted, the direction of stylus movement to the current position from the previous position (x_i, y_i) is quantized to one of the eight directions shown in Fig. 1. Figure 2 shows how the direction of the first element of a "segment"--i.e., a set of contiguous points having the same direction--is determined. In Fig. 2, this initial direction is 90°. Figure 3 illustrates how the directions of subsequent elements in the segment are determined. Figure 3 also shows how the band around the initial direction has been

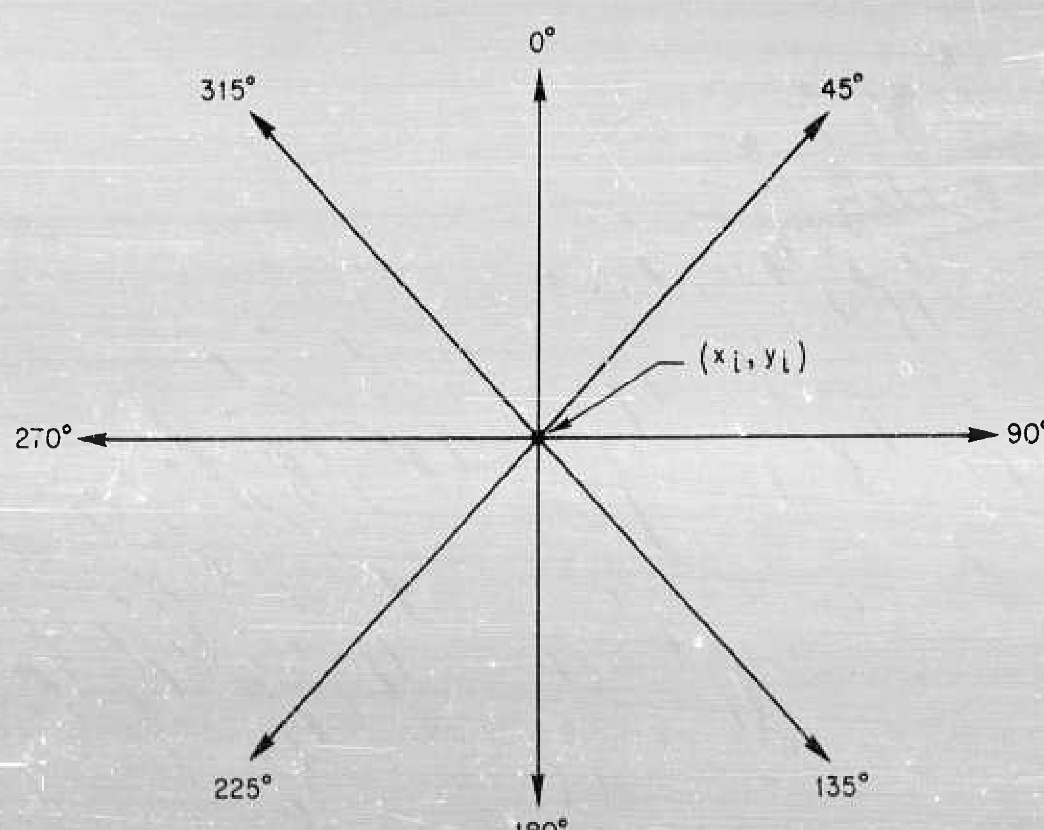

Fig. 1—The 8 directions to which the track is quantized

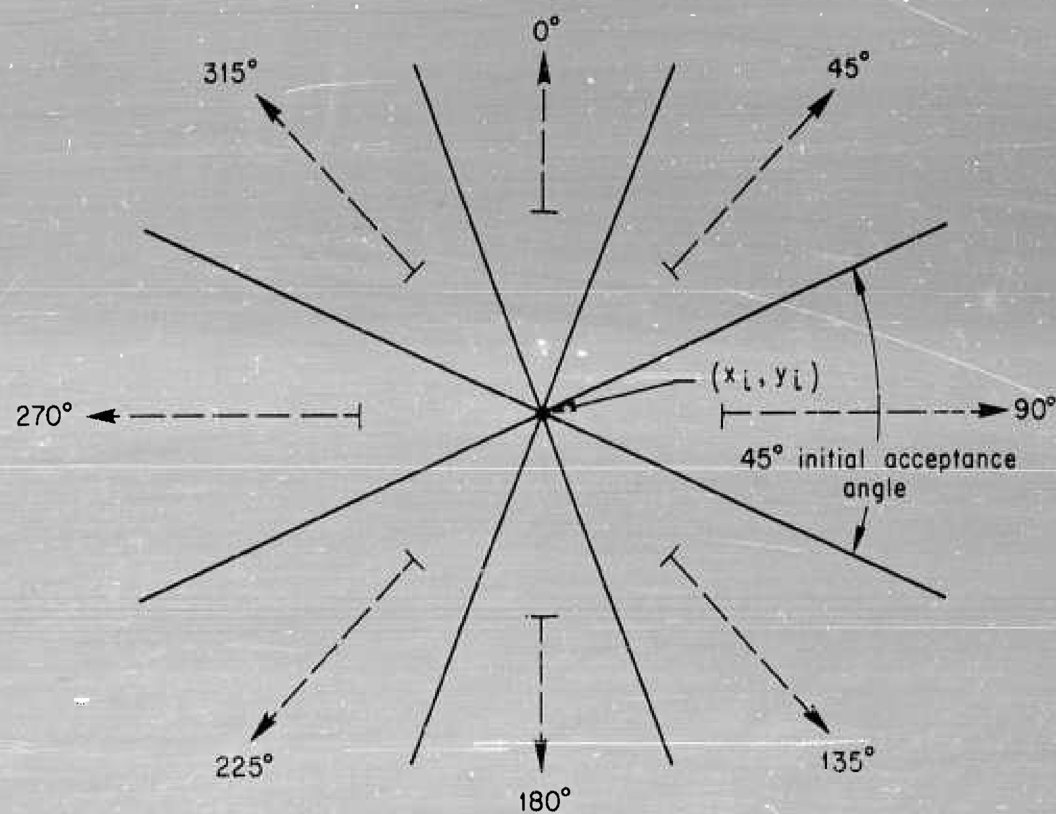

Fig. 2—Initial criteria for direction quantization

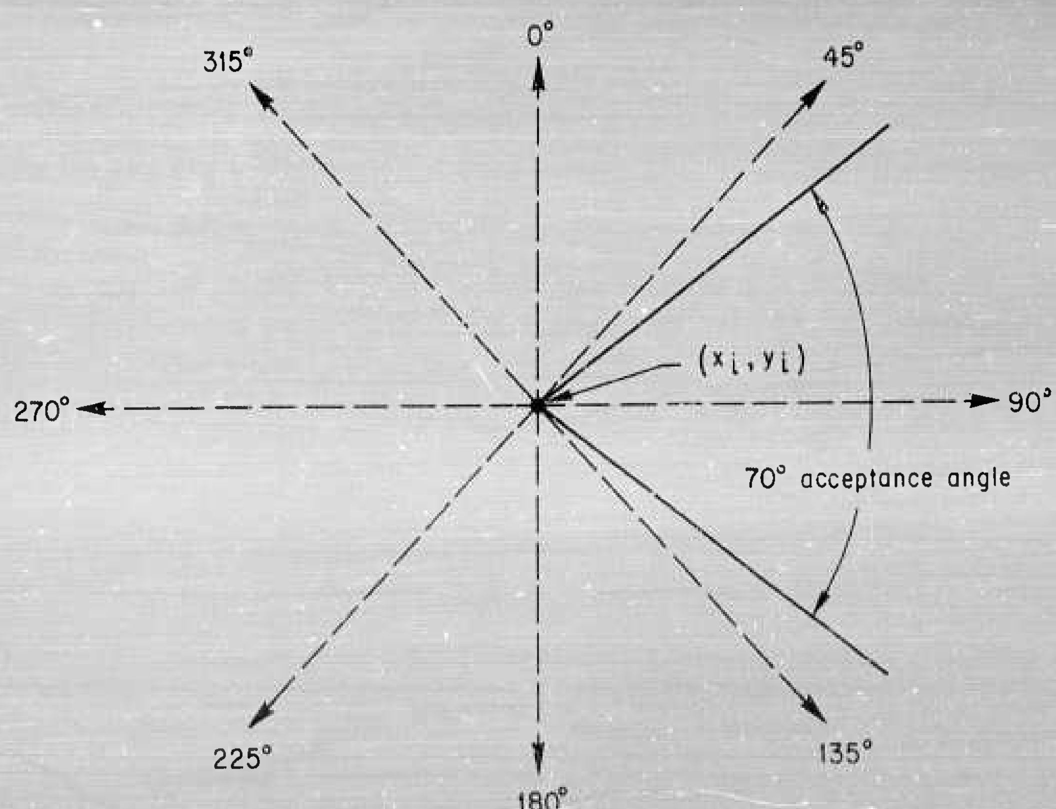

Fig. 3—Hysteresis enlargement of accepted direction criteria

widened from 45° to 70° to provide for some bias or hysteresis in direction assignment. This gives smoother segments and results only in a penalty to adjacent directions during change of direction, which in reality is the desired effect. It should be noted that in Figs. 2 and 3 the dotted lines indicate the assigned directions for directions actually falling within the solid boundaries.

Segments are formed as directions are assigned. Thus, a stroke can be characterized as a set of connected segments, each of which requires two parameters, d and n, to describe it; d is the direction of the segment and n is the number of elements in the segment. It should be noted that by virtue of the smoothing and filtering algorithm used, diagonal segments may be as much as 1.4 times as long as those on the vertical or horizontal axes.

As an example, let us take the stroke for the character we call "two." In Fig. 4 the solid line represents the actual path of the stylus, while the dots denote the smoothed and filtered point coordinates.

Figure 5 shows a representation (which is never displayed) of the quantized-direction version of the character. Any stroke, S, appears in the computer as a description in the following form:

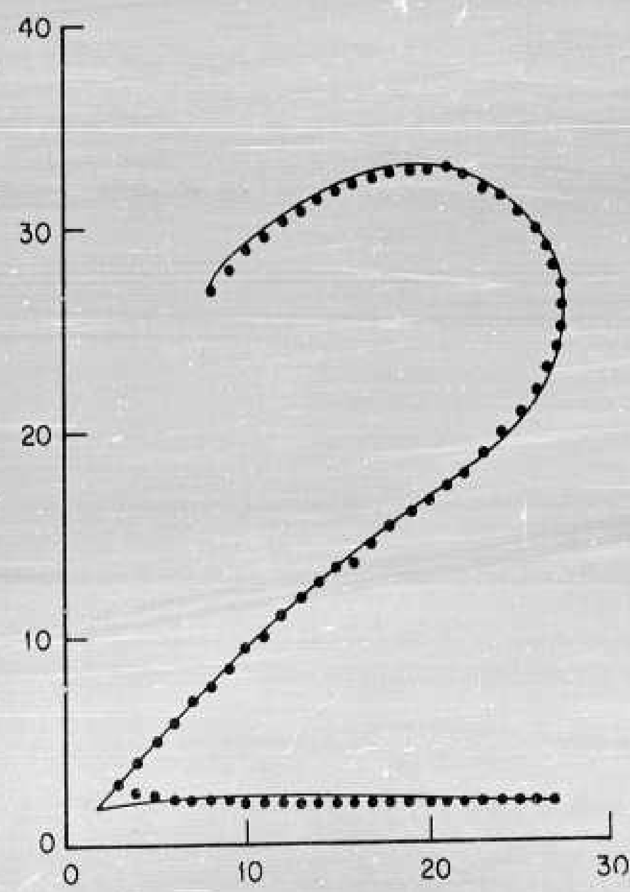

Fig. 4—Initial and smoothed track

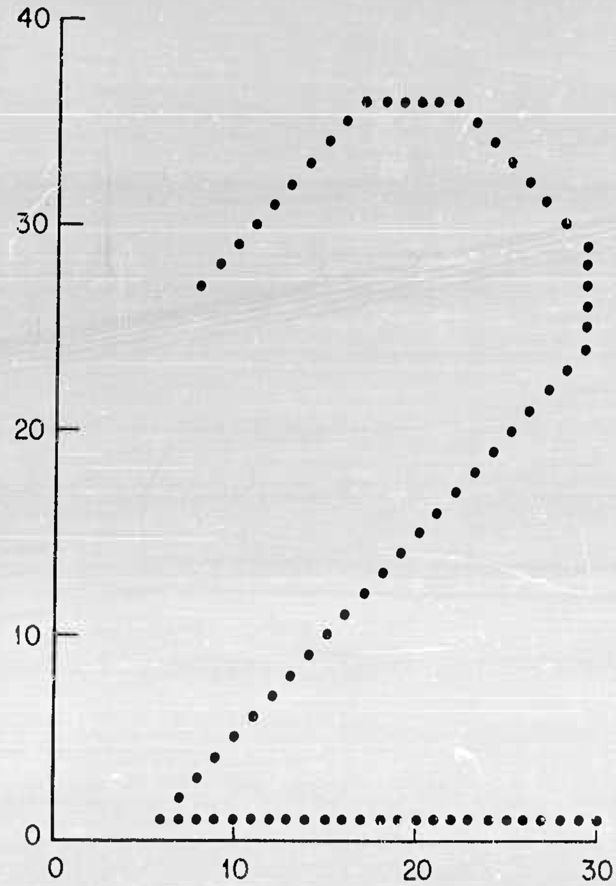

Fig. 5—Segmented track

$$S = (x_0,y_0)(d_1,n_1)(d_2,n_2) \ \ldots \ (d_k,n_k)(x_n,y_n) \qquad (2)$$

where (x_0,y_0) is the first point of the stroke, (x_n,y_n) is the last accepted point of the stroke, and the (d_i,n_i) are the segment descriptions. In the case of our "two," we have:

$$S(\text{"two"}) = (8,27)(45^\circ,9)(90^\circ,5)(135^\circ,7)(180^\circ,5)$$
$$(225^\circ,23)(90^\circ,24)(27,2) \ . \qquad (2a)$$

IV. DESCRIPTIONS AND FEATURES

Because angular measure is not the most convenient one for computer manipulation, the following values have been substituted for the angular directions:

$$0^\circ \rightarrow 0$$
$$45^\circ \rightarrow 1/4$$
$$90^\circ \rightarrow 1/2$$
$$135^\circ \rightarrow 3/4$$
$$180^\circ \rightarrow -1$$
$$225^\circ \rightarrow -3/4$$
$$270^\circ \rightarrow -1/2$$
$$315^\circ \rightarrow -1/4$$

These values were chosen because the amount of "turning" and the direction of the turning (clockwise or anti-clockwise) between adjacent segments can be computed directly, using three-bit binary two's complement arithmetic. By this method, turns are limited to a maximum of 180° in either direction. Differencing the values of adjacent segment directions generates a "navigational" description of the stroke. For the example of the "two," this becomes

$$S'(\text{"two"}) = (8,27)(1/4,9;\ 1/4,5;\ 1/4,7;\ 1/4,5;\ 1/4,23;\ -3/4,24)(27,2) \quad (3)$$

which says: beginning at (x_0,y_0), take a heading of 45^o for nine units; then turn clockwise 45^o and proceed 5 units; ...; and finally, turn anti-clockwise 135^o and proceed 24 units (which is how Fig. 5 was drawn). It is highly unlikely that either (2) or (3) will take one from coordinates (8,27), via the exact prescribed path, and end up precisely at (27,2). Therefore, the (x_n,y_n) is superfluous. Further, by eliminating (x_0,y_0) and the initial direction, S' can be transformed into a position- and rotation-independent description. And, if the segment lengths are eliminated, we make the description size-free. In essence, then, we have created a size-, position-, and rotation-independent description of a stroke. In so doing, we have introduced some distortion, but hopefully not so much as to destroy the basic qualities required for recognition. Notice that a stroke which contains no curvature now has no description--i.e., it is "empty."

The description of our "two" now becomes

$$S''(\text{"two"}) = (1/4,\ 1/4,\ 1/4,\ 1/4,\ -3/4)\ . \quad (4)$$

To get a feeling for how much distortion has been introduced, Fig. 6 shows the above description (assuming an initial heading of 45°) drawn with unit segments.

The next step is the transformation of the navigational description of the stroke to one containing "features."

The feature table, Fig. 7, is a double-entry table giving a unique "feature" for each pair of parameters, T and k. This, in essence, is the only way in which features are defined. The division or selection of which figures are to be grouped as a feature was arbitrary and based upon experience and intuition.

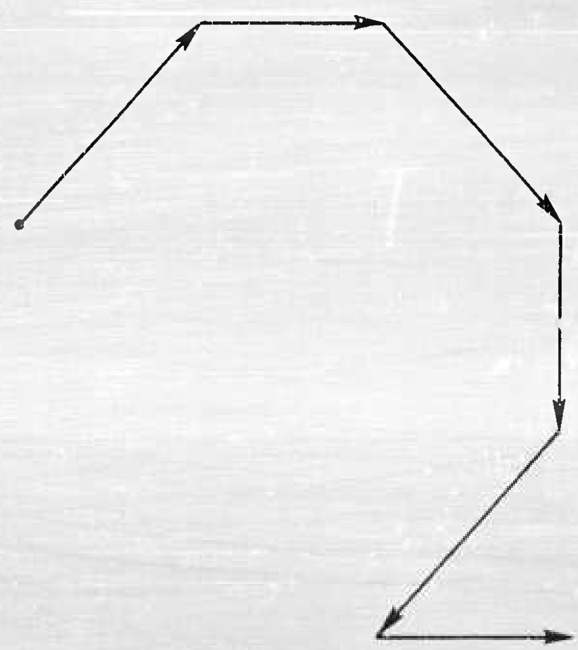

Fig. 6—Literal interpretation of segment encoding

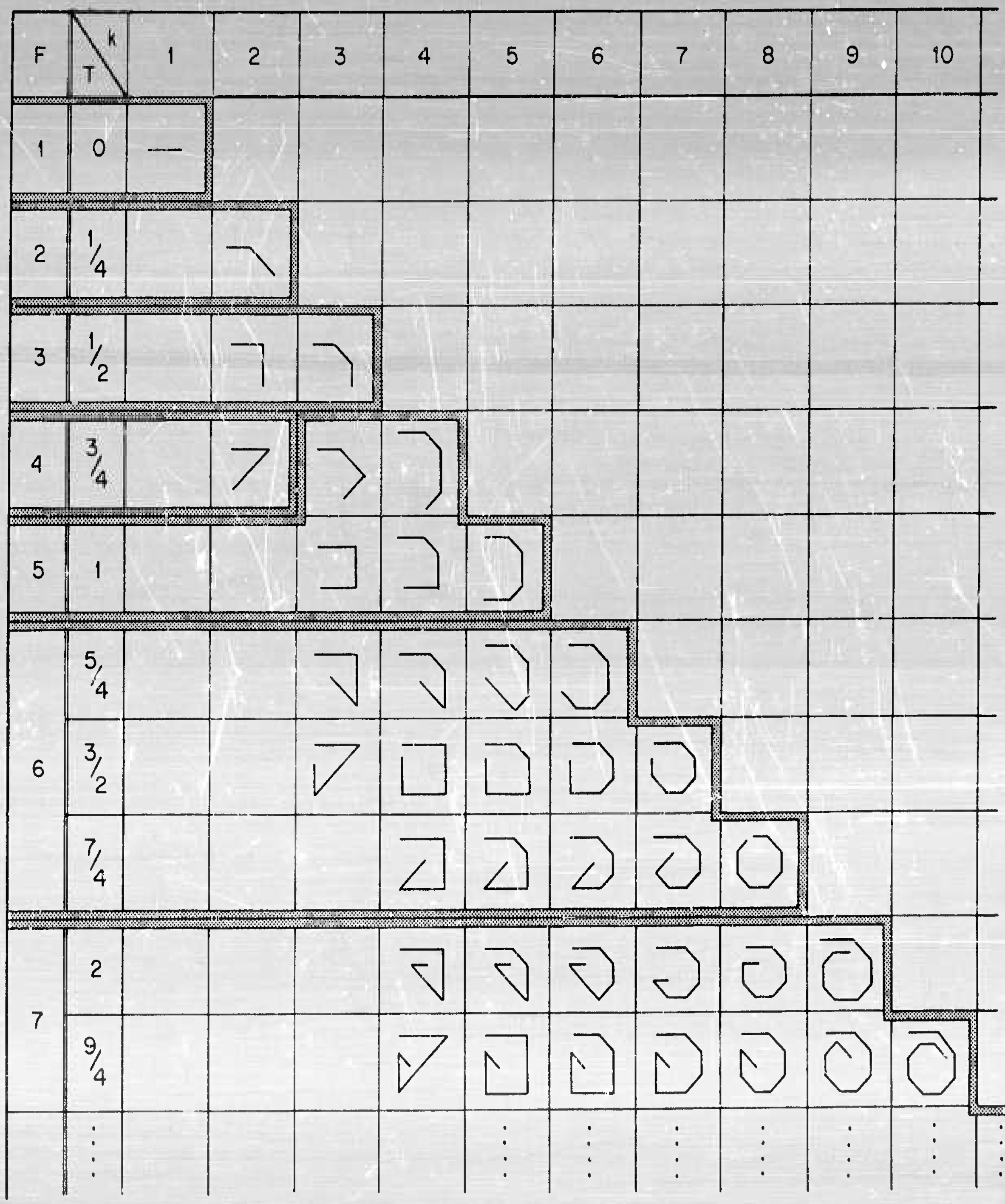

Fig. 7—Feature table

The entries in the body of the feature table consist of single illustrative examples of all the possible figures that could occur having a particular value for T and k. The heavy lines indicate the boundaries of the various feature categories. Blank entries indicate that no figure can exist (within the rules used here) for that T,k combination.

The procedure for extracting features from the description S" is a simple one. Beginning with the first segment of S", items of the form (T,k,r_i) are formed (where T is the sum of the values of adjacent segments of the same sign, k is the number of segments in the sum, and r_i is the next segment--the one for which the sign change was detected). The value of r_i is not included in the formation of the subsequent T, but is counted in the subsequent k. In the final item, r_i is vacuous. The feature table is entered with the arguments $(|T|,k)$; the resulting feature $F(|T|,k)$, with the sign of T affixed, replaces the pair T,k in the item.

Applying this procedure to the description of the "two" in (4), we get

$$S'''(\text{two}) = (1,4,-3/4;\ 0,1,-) = (F_5,-3/4;\ F_1,-) \qquad (5)$$

which says that the stroke is described as a "feature" of type 5, drawn clockwise, connected to a "feature" of type 1 by an anti-clockwise rotation of 135°. Figure 8 shows three possible "twos" which have the same description.

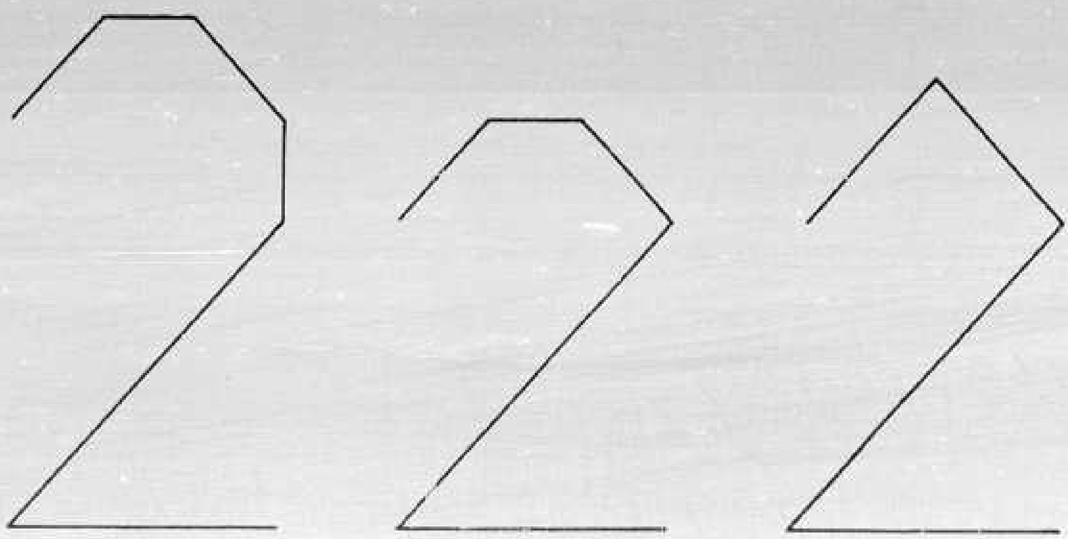

Fig. 8—Segmented forms of differently drawn 2's which have the same description

This last transformation of the stroke description contains what appears to be the minimum necessary information for recognizing characters, but it is not sufficient. The most important consideration lacking is the relation between the beginning and end of the stroke--the information which permits discrimination between "zero," "six," and "nine"; "one," "slash," and "minus"; etc.

V. END-POINT RELATIONS

Given the actual end-point coordinates of a stroke, (x_0,y_0) and (x_n,y_n), a position- and size-independent measure for them can be generated using the same technique as in the direction or heading assignment, with one added fillip. Before determining the quantized direction from (x_0,y_0) to (x_n,y_n), it must first be determined whether or not the two end-points are "coincident." Coincidence is defined as the case where (x_n,y_n) lies within some computed distance of (x_0,y_0). The best measure that has proven useful is to take some fraction (1/m) of the stroke lengths (the number of points taken for the stroke) as the basis for the computation. Three geometric relations are available:

$$|x_n-x_0| \text{ or } |y_n-y_0| \le s/m \tag{6}$$

$$(x_n-x_0)^2 + (y_n-y_0)^2 \le (s/m)^2 \tag{7}$$

$$|x_n-x_0| + |y_n-y_0| \le s/m \; . \tag{8}$$

Relation (6) describes a square of side 2s/m about (x_0,y_0), within which (x_n,y_n) must fall to be considered coincident;

relation (7) is a circle of radius s/m; and (8) is a diamond with height s/m. Relation (6) was chosen for reasons of simplicity and compatibility with the smoothing algorithm.

To the description of the stroke, then, we now add the end-point relation, which restores, in some instances, the attitude of the stroke, making it no longer completely rotation-independent. When the end-points are coincident, though, the stroke description remains rotation-independent.

The following notations will be used for the nine end-point relations:

0 : (x_0, y_0) and (x_n, y_n) are coincident

↑ : (x_n, y_n) is "directly" above (x_0, y_0)

↗ : (x_n, y_n) is above and to the right of (x_0, y_0)

→ : (x_n, y_n) is to the right of (x_0, y_0)

↘ : (x_n, y_n) is below and to the right of (x_0, y_0)

↓ : (x_n, y_n) is "directly" below (x_0, y_0)

↙ : (x_n, y_n) is below and to the left of (x_0, y_0)

← : (x_n, y_n) is to the left of (x_0, y_0)

↖ : (x_n, y_n) is above and to the left of (x_0, y_0).

Now, the completed description of the "two" becomes:

$$S'''(\text{two}) = (F_5,\ -3/4,\ F_1, \searrow)\ . \tag{9}$$

VI. MULTI-STROKE FIGURES

Each stroke description of a multi-stroke figure must be tied to the next stroke so that all the strokes belonging to a particular figure are connected together in the order in which they are drawn. (How one determines which of a series of consecutive strokes belong to one figure and which to the next will not be discussed here.)

Tying together the individual stroke descriptions into a figure description is accomplished by replacing the vacuous connective at the end of a stroke with the connective "$\wedge$" and appending the subsequent stroke description.

Using this notation, a two-stroke "five" may have the description:

$$S'''(\text{five}) = (-F_3, 1/4;\ F_5, \wedge;\ F_1, -)\ . \qquad (10)$$

In this case, $-F_3$ indicates that the feature F_3 was drawn in an anti-clockwise direction.

End-point relations for multi-stroke figures are a bit more of a problem than for single-stroke figures.

There seems to be no reasonable subset of all possible end-point relations that does not lead to ambiguity or confusion in description; therefore, all end-point relations are computed and added to the description. For this reason, in all the work done using this method, figures were limited to a maximum of four strokes--the number of end-point relations per figure being

$$N(2N-1) \tag{11}$$

where N is the number of strokes in a figure. Thus, for a one-stroke figure we have one relation; for two strokes, 6; for three, 15; and for four, 28. At four bits per relation, four-stroke characters require 112 bits to describe the end-point relations alone.

The relations are computed in an orderly fashion and appended to the stroke descriptions. The following list of comparisons shows the order and number made, where B_i is the beginning of stroke i, E_i is its end, $\bar{c}$ stands for "compared with", and the comma (,) stands for "and".

One-stroke figures:

$B_1 \bar{c} E_1$.

Two-stroke figures:

$B_1 \bar{c} E_1, B_2, E_2;$

$E_1 \bar{c} B_2, E_2;$

$B_2 \bar{c} E_2$.

Three-stroke figures:

$B_1 \bar{c} E_1, B_2, E_3, B_3, E_3;$

$E_1 \bar{c} B_2, E_2, B_3, E_3;$

$B_2 \bar{c} E_2, B_3, E_3;$

$E_2 \bar{c} B_3, E_3;$

$B_3 \bar{c} E_3$.

Four-stroke figures:

$B_1 \bar{c} E_1, B_2, E_2, B_3, E_3, B_4, E_4;$

$E_1 \bar{c} B_2, E_2, B_3, E_3, B_4, E_4;$

$B_2 \bar{c} E_2, B_3, E_3, B_4, E_4;$

$E_2 \bar{c} B_3, E_3, B_4, E_4;$

$B_3\bar{c}E_3, B_4, E_4;$

$E_3\bar{c}B_4, E_4;$

$B_4\bar{c}E_4$.

For figures made up entirely of straight-line strokes (i.e., "featureless"), the end-point relations provide a unique mechanism which is position- and size-independent.

The "complete" description, then, of our two-stroke "five" becomes:

$$S'''(\text{five}) = (-F_3, 1/4;\ F_5, \wedge;\ F_1, -;\ \downarrow, 0, \rightarrow, \uparrow, \nearrow, \rightarrow)\ . \qquad (12)$$

VII. CONCLUSIONS

Tests run on a program incorporating basically the technique described in this Memorandum produced some interesting results. Before continuing, though, it should be noted that the test procedure was, for various reasons, far from exhaustive and well-controlled. Therefore, any figures quoted will be based more on personal evaluation than on statistical analysis.

The testing technique was a relatively simple one in which the user (usually the author) was permitted to "condition" the program with what he considered an appropriate set of samples for the test alphabet, and then to test the system with single characters, returning to the "conditioning" mode as he saw fit. On the average, for an alphabet of between 50 and 60 characters, it was necessary for the user to supply between 100 and 150 samples.

The technique worked best (as might be suspected) on "simple," single-stroke characters. Correct recognition rate was better for one-stroke than for two-stroke figures, which in turn was better than the recognition rate for three-stroke characters, etc. The conclusion (supported

by examination of description and samples) drawn from this is that the end-point analysis is far too discriminating, making minor differences predominant.

With multi-stroke characters, figures containing only vertical and horizontal strokes were more easily recognized than those containing slanted strokes. There are two reasons for this interesting aspect. The first is that what the user's eye calls slanted and what the program called slanted were not always in one-to-one correspondence. The second reason (and as yet not quite fully understood) is that the end-point analysis was more discriminating on slanted strokes than on vertical or horizontal. This could be due to an uncorrected coding error, an implementation error, or a conceptual error.

A way should be found to retain the kind of information contained in the end-point analysis, but without the current high penalty of extreme discrimination. To bring performance for a system based upon this technique up to an acceptable recognition level (above 95 per cent), it seems necessary to develop a method for determining the best approximation when an exact match does not exist. A third, and very important, capability which must be incorporated in a dynamic system (and which the above

method ignores) is the collection of all strokes belonging to the figure currently under examination without acquiring some of the strokes for the next character--for example, one must be able to discriminate dynamically between "13" and "B" in "real time."

No definitive conclusions can be drawn from this particular model about the applicability and limitations of this method. However, results have been encouraging enough to stimulate interest in incorporating its fundamentals in another model in which solutions to stroke-association and end-point discrimination problems will be attempted.

REFERENCES

1. Davis, M. R., and T. O. Ellis, The RAND Tablet: A Man-Machine Graphical Communication Device, The RAND Corporation, RM-4122-ARPA, August 1964.

2. Dimond, T. L., "Devices for Reading Handwritten Characters," Proceedings of the Eastern Joint Computer Conference (1957), Institute of Radio Engineers, Inc., New York, 1958, pp. 232-237.

3. Tietelman, W., New Methods for Real-Time Recognition of Hand-Drawn Characters, Bolt, Beranek, and Newman, Inc., Report No. 1015, June 1963.

UNCLASSIFIED

UNCLASSIFIED